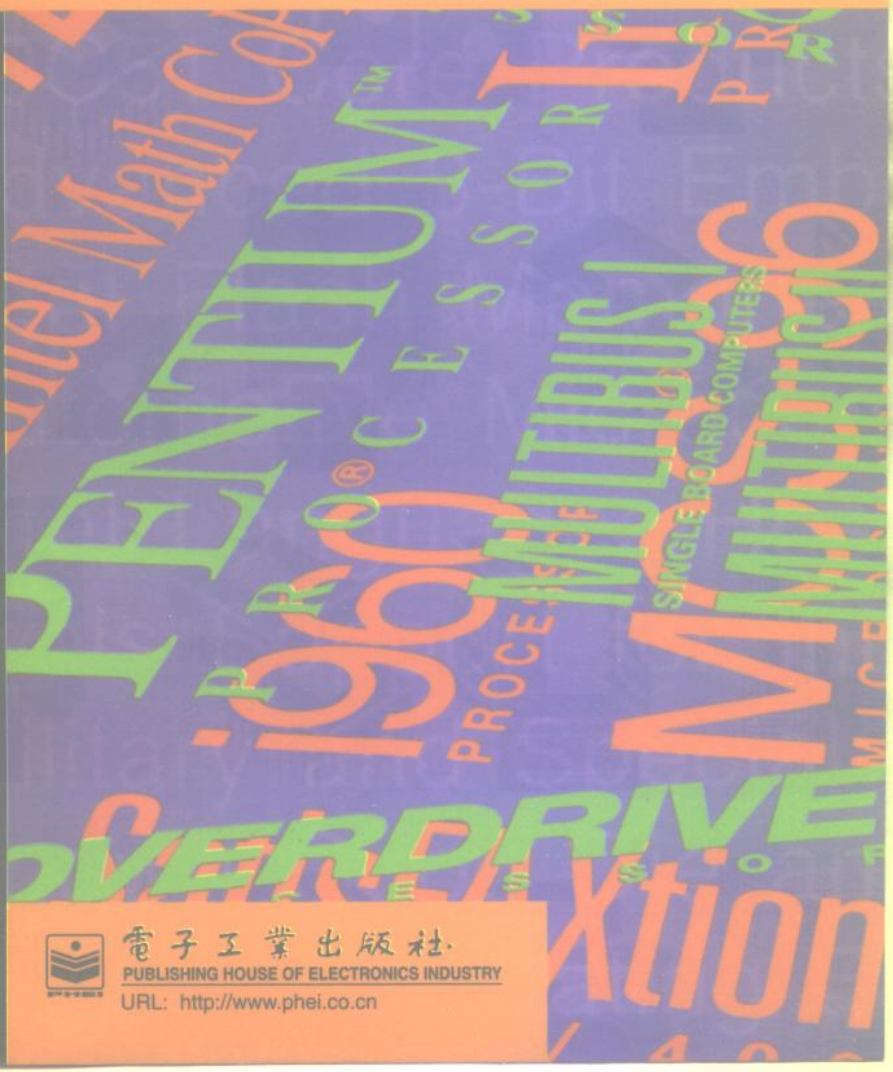
32位
系统软件编程指南

程荷 武航 译



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: http://www.phei.co.cn

intel

32位系统软件编程指南

[美] Intel公司 著

程荷、武航 译

電子工業出版社

Publishing House of Electronics Industry

内容简介

本书描述了80386系统结构和低级操作系统机构间的接口。详细地描述了任务和存储器管理，中断及与其紧密相关的异常，操作系统调用的实现，I/O功能，怎样进入保护的32位模式，与协处理器交互，运行80286及8086软件和实现UNIX System V的方法等内容。

本书结构严谨，层次分明，可作为编写80386微处理器操作系统、系统软件和底层软件的程序员的必备参考书。

书　　名：32位系统软件编程指南

著　　者：[美]Intel公司　著

译　　者：程荷、武航

责任编辑：邓露林

印 刷 者：北京牛山世兴印刷厂

装 订 者：三河市路通装订厂

出版发行：电子工业出版社出版、发行

北京市海淀区万寿路173信箱　邮编100036　发行部电话68214070

URL：http://www.phei.co.cn

经　　销：各地新华书店经销

开　　本：850×1168 1/32 印张：5.375　字数：143千字

版　　次：1997年3月第一版　1997年3月第一次印刷

印　　数：5000册

书　　号：ISBN 7-5053-3942-7/TP·1710

定　　价：8.00元

著作权合同登记号　图字：01-96-0876

前　言

32 位系统软件编程指南描述了 80386 系统结构和低级操作系统机构间的接口。它不讨论操作系统的策略问题或操作系统独立于一个处理器结构的功能。例如，这本书展示了一个操作系统是怎样使用 80386 的任务转换指令调度一个新任务(进程)的，但它却不讨论操作系统选择要调度的任务所能够采用的许多策略。为了引证另一个例子，32 位系统软件编程指南包含了 80386 设备的输入/输出功能，但却把文件 I/O 的讨论留给操作系统的教科书。

读　者

这本书主要是为正在研制 80386 微处理器操作系统的系统程序员所写。正在编写其它系统软件，如链接程序和实用程序的程序员也可以从本书获益。本书也可能对想要了解 80386 的结构设施是怎样支持通常的操作系统机构的任何人都是有价值的。

为了成功地使用这本书，你必须充分地熟悉多任务操作系统。

相关出版物

32 位系统软件编程指南是描述 80386 微处理器的四本 Intel 出版物中的一本。其它几个出版物是：

- 80386 入门，订购号为 231252
- 80386 程序员参考手册，订购号为 230985
- 80386 硬件参考手册，订购号为 231298

阅读 32 位系统软件编程指南可以独立于 80386 硬件参考手册。80386 入门是必不可缺的，而 80386 程序员参考手册是这本书

的姊妹篇之一。

在阅读本书之前，你应当彻底地理解 80386 入门中的资料，特别是第二章（应用结构）和第三章（系统结构）。如果你对在 80386 上运行 8086 或 80286 的程序感兴趣，也需要阅读第四章（结构兼容性）。在阅读本书之前，你应当浏览 80386 程序员参考手册，在阅读 32 位系统软件编程指南时，应把它保持在手边。32 位系统软件编程指南经常简化结构特征的描述，以便更清楚地展示这些特征与操作系统机构的关系。当你需要任何 80386 功能的明确的描述符时，参考 80386 程序员参考手册。书中的一些例子是用 ASM 386，Intel 80386 汇编语言写的，这可参见 ASM 386 汇编语言参考手册。

怎样读这本书

32 位系统软件编程指南安排了十章的内容，最具特色的主题包含在本书的末尾。前七章描述了 80386 受保护的 32 位操作，此操作模式最有可能为新的 80386 应用所选择。使 80386 与早期的 Intel 86 系列处理器相兼容的特征描述在第八章和第九章中，而第十章描述了在 80386 上实现 UNIX System V 操作系统的一种方法。

第一章和第二章描述了任务和存储器管理。这些主题紧密相关，你会发现在第一章中经常要涉及到第二章。然而，阅读了 80386 入门后，大多数读者应当理解有关 80386 的存储器管理功能，达到足以不必再查阅前边那些参考资料的程度。第三章包含中断和与其紧密相关的异常。第四章描述了操作系统调用怎样在 80386 上实现。第五章描述了 80386 的输入/输出功能。前五章描述的 80386 好象早已运行在保护的 32 位模式中，并具有所有结构定义的合适的数据结构（如页表）。第六章告诉你怎样让 80386 从一个硬件 RESET 进入到保护的 32 位模式。

最后四章包含了特殊的主题，可以有选择地阅读。第七章描述

了 80386 和 80287、80387 数值处理器(或它们的软件仿真程序)之间的交互作用。第八章和第九章描述了为运行现存的 80286 和 8086 软件的 80386 的功能。最后一章是一个扩充的例子,它描述了在 80386 上实现 UNIX System V 的一种方法。

注意:本书中所给的代码例子没被测试过。

目　录

第一章 任 务

从功能上讲 80386 是一个多任务计算机。尽管处理器能被用在单任务系统中，但其系统结构的大多数设备都被设计成支持多任务的并发执行。例如，存储器管理，保护和异常处理都是基于任务的。80386 能完成操作系统的任务转换(关联转换)或自动响应一个中断或异常。本章描述操作系统能用来创建和管理任务的 80386 功能。与中断和异常相关的那些任务将在第三章中讲述。

1.1 任务执行环境

图 1-1 展示了 80386 的任务在执行期间可以使用的按结构定义的寄存器和数据结构。图 1-1 所示的大多数数据结构与中断处理和存储器管理的关系比任务管理更为密切，这在后面章节中描述。然而，任务状态段是任务管理的中心，它是这一章的主题。

1.2 任务状态段和描述符

任务状态段能被考虑成两部分：机器状态，主要由寄存器值组成，和软件状态，由文件描述符、调度参数和其它操作系统定义的数据组成。一个多任务操作系统按传统在一个“任务控制块”或一个被同样命名的记录(或记录集)中记录每个任务的机器状态和软件状态。

80386 系统结构定义了一个保持某任务机器状态的记录。这个记录称作任务状态段，如图 1-2 所示。操作系统初始化一个新任务的 TSS，而 80386 保存 TSS，在任务转换时读写它，在特权级变化时读取它。80386 仅指定 TSS 的前 26 个双字格式，可选择地直

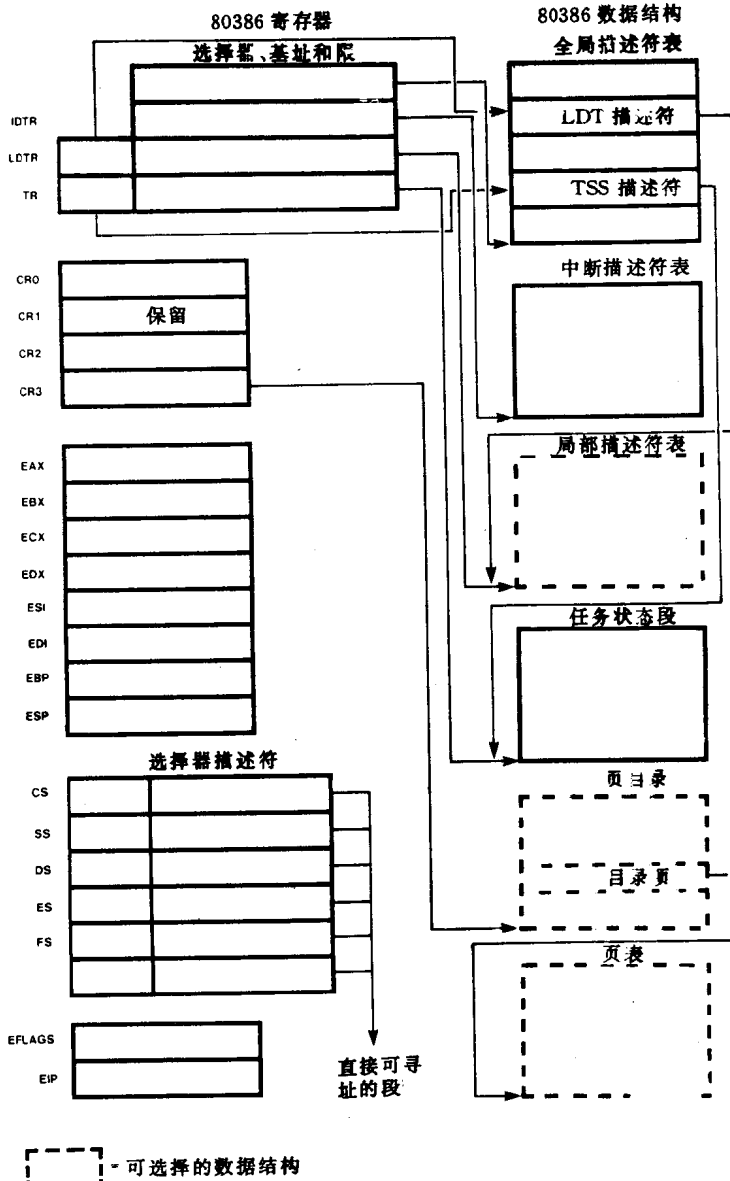

80386 寄存器
选择器、基址和限
IDTR
LDTR
TR
80386 数据结构
全局描述符表
LDT 描述符
TSS 描述符
中断描述符表
局部描述符表
CR0
CR1
保留
CR2
CR3
EAX
EBX
ECX
EDX
ESI
EDI
EBP
ESP
任务状态段
选择器描述符
CS
SS
DS
ES
FS
页目录
目录项
页表
EFLAGS
EIP
直接可寻址的段
可选择的数据结构

图 1-1　任务执行环境

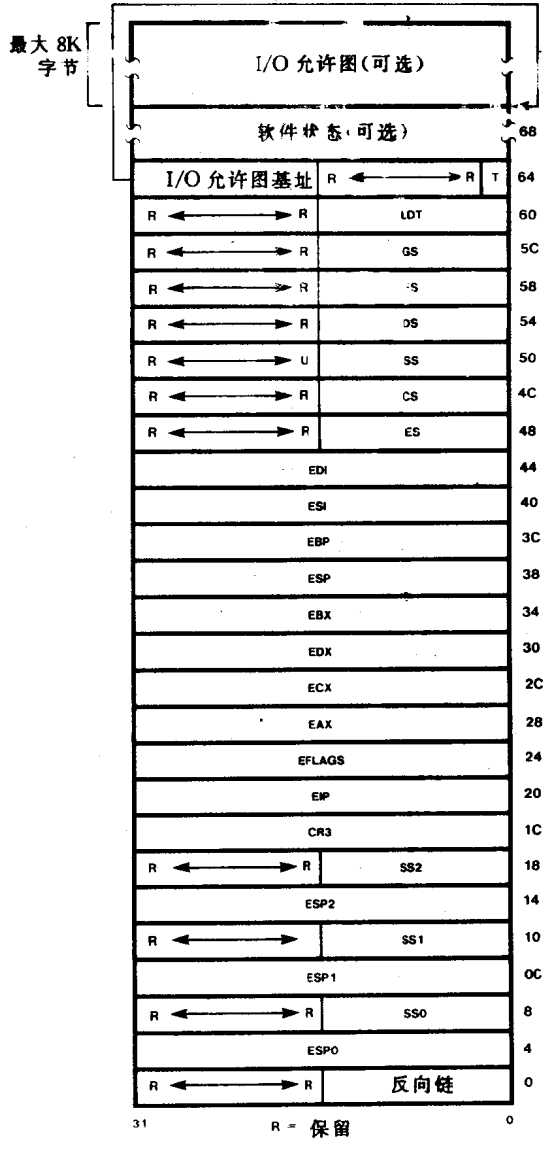
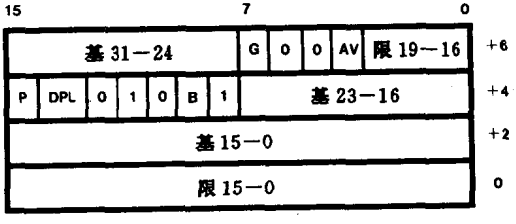

最大 8K
字节
I/O 允许图(可选)
软件状态(可选)
68
I/O 允许图基址
R
R
T
64
R
R
LDT
60
R
R
GS
5C
R
R
FS
58
R
R
DS
54
R
U
SS
50
R
R
CS
4C
R
R
ES
48
EDI
44
ESI
40
EBP
3C
ESP
38
EBX
34
EDX
30
ECX
2C
EAX
28
EFLAGS
24
EIP
20
CR3
1C
R
R
SS2
18
ESP2
14
R
SS1
10
ESP1
0C
R
R
SS0
8
ESP0
4
R
R
反向链
0
31
R = 保留
0

图 1-2　任务状态段

到后面的 8K 字节(64K I/O 地址空间的 I/O 位图)。操作系统可以自由地使用介于 I/O 位图和 TSS 内核(前 26 个双字)之间的区域来记录一个任务的软件状态。

由于一个 TSS 是 386 定义的一个段,因而它必须有一个描述符。图 1-3 展示了 80386 描述符的格式。其基址,界限,粒度,可使用,存在和描述符特权级域与它们的代码和数据段描述符的副本相同(这些在第二章讲述)。注意,如果这些域被操作系统定义和使用(I/O 允许图在第五章中讲述),TSS 的界限就必须考虑可选的 I/O 允许图和任务软件状态。如果 I/O 允许图不存在,界限就必须设置至少为 68H(机器状态数据长度);如果操作系统用软件状态信息来扩展一个 TSS,界限就能——但并不需要——覆盖附加的信息。TSS 必须驻留在全局描述符表(GDT)中,以便使处理器不管正在运行的任务(如第三章所述,中断和异常能触发任务转换)而访问所有的 TSS。为了防止越权的任务转换,TSS 的描述符应分配为特权级 0。



说明
G: 粒度
0=1 字节
1=4K 字节
AV: OS 可使用
P: 存在
0=TSS 不存在
1=TSS 存在
DPL: 描述符特权级
B: TSS 忙
0=可用 1=忙

图 1-3 任务状态段描述符

80386 在 TSS 描述符中设置了捕获企图递归调用一个任务的忙(B)位；操作系统应当初始化这一位为 0。在通常的任务转换中，80386 置位新任务的忙位并清除旧任务的忙位。然而，在嵌入式的任务转换中，80386 保留旧任务的忙位设置。一个嵌入式的任务转换发生在一个任务调用另一个任务的时候，更通常的是发生在 80386 调用作为一个被执行的任务(见第三章)的中断或异常处理程序的时候。调用忙位被置位的任务的企图会导致一个无效 TSS 异常。

1.3 任务创建

图 1-1 所示的 80386 定义的数据结构必须在转换到一个新任务之前设置好。GDT 和中断描述符表(IDT)是系统范围内的资源，它们能由 Intel System Builder 实用程序，或操作系统初始化时静态地创建，就像第六章所讨论的那样。必须为新任务创建一个新的局部描述符表(LDT)，除非这个新任务共享另一个任务的 LDT，或系统并不使用 LDT；相关任务和 LDT 的规则在第二章中讲述。如果允许分页的话，任务就需要一个页目录和一个或更多的页表(所有的任务都能共享单一的页目录和一套页表)。LDT、页目录以及页表的创建在第二章讨论。

一个操作系统不能靠直接写 TSS 和 GDT 来初始化一个 TSS 和 TSS 描述符，而必须使用数据段别名。别名的段在线性地址空间中是相互重叠的；在第二章中会进一步地描述。

当初始化 TSS 时，操作系统应当遵循下述准则：

- 反向链：这个域应当被初始化为 0，以防止由于错误的任务转换而错误地设置 NT(嵌入式任务)标志。如果一个任务的 NT 标志被置位，80386 执行 IRET 指令，就会转换到其选择器被记录在反向链域中的那个任务。当一个任务被中断或引起其处理程序是一个任务的异常，或当一个任务调用另一个任务时，80386 就设置 NT 位，并更新反向链

域。一个任务能用 POPF 指令设置它的 NT 位，但它只有对操作系统的 TSS 别名加以访问，才能更新它的反向链域。如果 NT 已被错误地置位，而任务发出 IRET 指令时，操作系统则可通过初始化反向链为 0 使 80386 发出一个无效的 TSS 故障。

- 特权堆栈指针：SS0、SS1、SS2、ESP0、ESP1 和 ESP2 必须包含 0～2 特权级各自的初始化堆栈选择器和偏移量。操作系统必须初始化由它或其它软件所使用的符合特权级的这些域。例如，一个操作系统运行特权级为 3 的用户代码，而它的特权级是 0，它就必须初始化 SS0 和 ESP0。作为系统调用、中断或异常的结果，当 80386 从特权级 3 变化到特权级 0 时，它就通过将 SS0 装入到 SS 段寄存器，ESP0 装入到 EPS 寄存器来转换特权堆栈。
- CR3：如果允许分页，TSS 的 CR3 域就必须用任务页目录的物理地址来初始化。
- EIP、EFLAGS、通用和段寄存器：初始化为任务开始运行时应具有的值。
- LDT：用任务 LDT 的选择器初始化；如果一个任务没有使用 LDT，这个域必须设置为 0(空选择器)。
- T 位：当处理器转换到这个任务时，通过设置这一位，操作系统能使 80386 产生调试陷阱(见第三章)。
- I/O 允许图基址和可选的 I/O 允许图：这些域能用来准许一个任务访问可选择的 I/O 端口(见第五章)。

1.4 任务终结

任务终结通常是操作系统设计问题，很少受 80386 系统结构的影响。典型地，终结处理是介于操作系统出口程序和系统回收任务之间的过程。运行在要终结的任务上下文中的特权级出口程序能直接访问任务的地址空间和软件状态。简而言之，出口程序将任